LA-UR-19-20673
Approved for public release; distribution is unlimited.

Title: User Guide for the New Tally ACODE Capabilities

Author(s): Josey, Colin James

Intended for: Report

Issued: 2019-01-29

User Guide for the New Tally ACODE Capabilities

Colin Josey

January 22, 2019

In the newest development version of MCNP (post-6.2.1), the α-eigenvalue solver has undergone some changes. A new approach for computing α was added alongside the current `ACODE` algorithm. This document will briefly discuss the changes made by this modification, both in the current `ACODE` and in the new algorithm, as well as the new capabilities this method provides.

The Method

The α-eigenvalue solution to the transport equation is created via a separation of variables. The time-dependent flux is replaced with

$$\psi(\mathbf{r}, \mathbf{\Omega}, E, t) = \psi(\mathbf{r}, \mathbf{\Omega}, E)e^{\alpha t}.$$

Simplifying the resulting equation results in the steady-state transport equation with an additional $(\alpha/v)\psi$ term. When α is positive, this term acts as an absorption term ("time absorption"). When α is negative, this acts as a source ("time source").

The current MCNP algorithm solves for α using the k–α method. A k-eigenvalue problem is run with this $(\alpha/v)\psi$ term. α is adjusted such that $k \to 1$, at which point the equation simplifies to the α-eigenvalue equation. Currently, α is updated every 2-4 generations using Eq. (1).

$$\alpha_{n+1} = \alpha_n + (k_n - k_{\text{target}}) \left(\alpha_n + \frac{k_n}{\tau} \right) \tag{1}$$

Here, k is a user-selected k-eigenvalue type (collision, track-length, absorption) and τ is the mean neutron lifetime.

The new algorithm (tally α) makes two changes. First, α is updated at every generation. Second, the α updater is replaced with the solution to the following equation, where $[\cdot]$ is a surface tally and $\langle\cdot\rangle$ is a volume tally.

$$\alpha \left\langle \frac{1}{v} \right\rangle = -\left[\text{leakage}\right] - \langle\text{total}\rangle + \langle\text{scatter}\rangle + \langle\text{prompt fission}\rangle + \sum_{i=1}^{N} \frac{\lambda_i}{\alpha + \lambda_i} \langle\text{delayed fission}\rangle$$

This algorithm has three advantages compared to the current algorithm. First, it improved the figure of merit by 10% to 90% on tested problems. Second, it reduced bias from running too few neutrons per iteration. In some cases this reduction was by nearly an order of magnitude. And third, it *optionally* supports computing the true α-eigenvalue with the contribution of the delayed precursors.

A much more extensive discussion about the algorithm and its implementation can be found in LA-UR-18-22738. One notable change between that document and the current implementation is the removal of the collision and combined estimators. This was done due to the statistical instability of the collision estimator in void and near-void conditions. In normal k-eigenvalue, a collision estimator works fine with voids, as all of the necessary tallies are zero in the void. For the α-eigenvalue, the $1/v$ tally is not zero in the void. While there are still "collisions" in the void due to time source and time absorption effects, as α approaches zero, the variance of these tallies will grow.

How to Use

Usage is simple. In the `ACODE` card, set `KALPHA` to 5. If you want to compute α with delayed precursors, simply enable the `TOTNU` card as well. Enabling the `TOTNU` card with `KALPHA` *not* set to 5 will result in the delayed ν being considered a part of prompt ν. This was done so as to not change MCNP's current `ACODE` algorithm.

Since the generations per batch is always set to one in the new method, the variable `KALREG` does nothing. It can be ignored. All of the remaining options operate identically to the current implementation. A summary of `ACODE` options are shown in Table 1.

Parameter	Description	Default
NSRCK	Number of source histories per generation	1000
RKK	Initial guess for k_{eff}	1.0
IKZ	Number of batches before α updates[a]	30
KCT	Total number of batches	IKZ + 100
MSRK	Number of source points for which storage will be allocated	Larger of 4500 or 2×NSRCK
KNRM	Normalization method for tallies: 0 = weight, 1 = histories	0
KALPHA	Estimation method for α: 1. Collision k for k–α iteration 2. Absorption k for k–α iteration 3. Track length k for k–α iteration 4. Differential operator perturbation 5. Tally-based α solve (NEW)	3
KALSAV	Number of batches before α is averaged and tallying begins	automatic [b]
KALREG	Batch to start ln-ln regression and reduce number of generations per batch from 4 to 2 (ignored if KALPHA = 5)	KALSAV + 2
MKRP	Maximum number of batches for which values are retained on MCTAL or RUNTPE files	6500
ALPHA	Initial guess for α, gen./shake	0.0
ALMIN	Minimum value permitted for α, gen./shake	0.0
KTARG	Target value for k_{eff}	1.0

Table 1: MCNP ACODE Parameters

[a] Other documents mention that this is the inactive cycles. This might lead to confusion in comparison to KCODE, in which tallies begin once inactive cycles end. For ACODE, tallies begin after KALSAV.

[b] The automatic feature of KALSAV only checks to see that k_{eff} is converged to within one standard deviation of KTARG. It performs no convergence tests on the source.

Recommendations

Based on results from LA-UR-18-22738 and LA-UR-18-24670, a few recommendations can be made for the correct use of `ACODE` with the new modifications. Recommendations for each option of the `ACODE` card are listed below.

`NSRCK` - Number of source histories per generation

The number of source particles should be large enough such that the bias of the α distribution is acceptable. There are two cases in which one would need more neutrons per batch than commonly anticipated. The first case involves any simulation near critical. For these, the relative magnitude of the bias as compared to α can grow quite large. In extreme cases, this can prevent knowing if a geometry is subcritical or supercritical.

The second case involves the prompt critical boundary when delayed precursors are considered. Any simulation in which α can be on both sides of this boundary will have α heavily biased positive. In LA-UR-18-24670, the density of Jezebel was adjusted from subcritical to prompt supercritical. Even with 10 million particles per batch, biases in α of 3 orders of magnitude could be observed near the prompt supercritical threshold.

There are a few tests one can use to detect and estimate bias in α. One could examine the skew and excess kurtosis of the distribution. If either of these values substantially diverges from zero, it indicates that the distribution might not be normally distributed. Another test involves the median. The median often has a lower bias but higher variance as compared to the mean. As such, the difference between the two can be used to estimate the bias. The skew, excess kurtosis, and median are all output with the new tally `ACODE`.

`RKK` - Initial guess for $k_{\mathbf{eff}}$

This scaling parameter is only needed if reactor is initially very distant from critical. A large value will reduce the number of particles written to the fission bank in the first batch, possibly preventing crashes. Later batches will use the calculated k value and not `RKK`.

`IKZ` - Number of batches before α updates

A sufficient amount of time is needed between `IKZ` and `KALSAV` for both α and the source distribution to stabilize. There appears to be nearly no reason to set `IKZ` to a large value. As such, a value of 5 to 50 is adequate.

`KCT` - Total number of batches

The value `KCT` - `KALSAV` determines the number of active batches. This needs to be large enough that the distribution of batches can approach Gaussian. As such, this should be at least 50 larger than `KALSAV`, preferably more.

`MSRK` - Number of source points for which storage will be allocated

When simulating a model with a large negative α, increasing this value might improve stability and prevent source bank overruns. However, there are some simulations in which the time source can grow without bounds, and no value of `MSRK` will prevent crashes.

`KNRM` - Normalization method for tallies

This setting should be left as default.

`KALPHA` - Estimation method for α

The new tally algorithm outperforms the old one on all problems tested, so it is suggested to enable the new algorithm via `KALPHA` = 5.

`KALSAV` - Number of cycles before α is averaged and tallying begins

This should be large enough that both α and the source are converged. It is recommended to run the simulation with a smaller `NSRCK` and observe the Shannon entropy. If the Shannon entropy appears converged at iteration N, set `KALSAV` a bit higher than N. Upon completion of the new simulation, examine the Shannon entropy again to determine if `KALSAV` was sufficiently large. This should not be set to automatic, as it only checks for k convergence and will ignore source convergence.

KALREG - Batch to reduce internal settling generations and start ln-ln regression

This setting should be left as default.

MKRP - Number of batches for which data is retained

In order to use some of the diagnostics at the end of the simulation, such as the median, MKRP must be larger than KCT. This allows all of the α values to be available at once.

ALPHA - Initial guess

Setting ALPHA to the expected value of α might improve convergence. However, it is not necessary.

ALMIN - Minimum α

Some capabilities are incompatible with negative α, such as DXTRAN spheres. In addition, analog capture may give zero weights with α negative. If this is a concern, ALMIN should be set to zero. Otherwise, there are no ill effects to setting ALMIN too low. Conversely, if ALMIN is too large (such that a batch α needs resetting), it might bias the answer. For delayed neutron simulations, the batch α will never go below $(-\min\lambda)$, so any ALMIN below that value will yield identical results.

KTARG - Target $k_{\mathbf{eff}}$

If the k_{eff} of the geometry is known to be a certain value, but the computed k_{eff} is different, this can be used to adjust the problem closer to the right answer. It is important to note that this is a coarse approximation, as the discrepancy is almost certainly not due to just a proportionality error in the fission data.

The Output

The output of KALPHA = 5 is slightly different than the original algorithm. Alongside the track-length estimator, the skew and excess kurtosis is printed out at each iteration in order to diagnose bias. At the end of the simulation,

the median is computed and compared to the distribution. Finally, `mcplot` has been modified to properly plot values for tally α. The plots differ as compared to normal `ACODE`. The `kcode x` command for x = 1-6, 11-16, 18-30 will plot an assortment of values from the simulation. For $x < 20$, the plots are identical to normal `KCODE` runs. The added 11 plots are:

x	x Axis	y Axis
20	Cycle Number	Imposed α
21	Cycle Number	Calculated α
22	Cycle Number	Average Calculated α With Errorbars
23	Cycle Number	Std. Dev. of Mean of α
24	Cycle Number	Skew of Distribution of α
25	Cycle Number	Excess Kurtosis of Distribution of α
26	α	Frequency of Imposed α
27	α	Frequency of Calculated α
28	k	Frequency of Collision k
29	k	Frequency of Absorption k
30	k	Frequency of Track-Length k

Note that the imposed α is simply the calculated α from the previous step.

Changes to the Old `ACODE`

While no changes were made to the old `ACODE` algorithm itself (and as a result, transport will be performed exactly the same), a few changes were made to the output in order to make it more consistent. The changes are as follows:

- Shannon entropy is always output
- All α values are shown in scientific notation
- The confidence intervals for the combined k are now properly computed
- Minor spacing changes

These affect `outp` files as well as the terminal output.

Known Issues

There are a few known issues with the new algorithm. Most of them are either in the process of being fixed or are fixable depending on user needs. They are listed here for reference.

It is well-known that it is a challenge to simulate negative α-eigenvalues. The time source can grow without bounds and eventually cause the simulation to terminate. The new algorithm does not fix this, but it has been observed to occur less often. A future modification to MCNP is planned that will address this issue directly. As delayed precursors set a lower bound on α (and, as a result, a maximum value on the time source term), it is extremely unlikely for such a failure to occur when delayed precursors are considered.

For some nuclides at high energy (e.g., ^{30}Si above 20 MeV, ENDF/B-VII.1), the cross section data is not separated into distinct reactions. The contributions of many reactions are merged into "(n, anything)". The tally α algorithm needs to compute the excess neutron production at each collision, and does so through track-length tallies. An (n, anything) reaction would require a different tally approach. Since the flux at such high energies in an α calculation is extremely low, it is not expected that adding in this contribution will affect the answer. However, this feature can be added if it is found necessary.

Non-neutron paths for generating neutrons are not currently considered in computing α. As an example, if a neutron generates a photon, which generates a neutron, this will be ignored in tally α. Due to this, and the fact that tally `ACODE` is not thoroughly tested with non-`mode n` problems, an error message will appear if other particles are selected.

The tallied value of ν during calculations with delayed precursor is the "time-scaled" ν:

$$\nu(\alpha, E) = \nu_p(E) + \sum_{i=1}^{N} \frac{\lambda_i}{\alpha + \lambda_i} \nu_{di}(E)$$

Physically, this value has the meaning of

$$\nu(\alpha, E) = \frac{\text{number of } n \text{ appearing at time } t\text{, including delayed } n}{\text{fissions at time } t},$$

and is slightly different than the usual interpretation of

$$\nu(E) = \frac{\text{total number of } n \text{ generated from fissions at time } t}{\text{fissions at time } t}.$$

In negative α situations, $\nu(\alpha, E)$ will be larger, which shows the effect of a larger past neutron population coupled with decay. Conversely, in positive α situations, the past population is smaller than the current one, and so the delayed neutrons become less and less relevant. If there is a need, both ν could be added to the tally system. This distinction does only appears when delayed precursors are considered.